SYSTEM AND METHOD FOR HIGH SPEED, HIGH VOLUME TABULATION OF DATA

Field of the Invention

This invention relates to tabulation of data in a computer environment, and in particular to a system and method for tabulation of data in relatively high volumes and at relatively high speeds.

Background of the Invention

Data tabulation is a common data analysis tool for many statistical agencies. Tabulation of large a quantity of data, such as a population census, is often done through the use of a mainframe computer with results thereof formatted using personal computers or the like.

The use of a mainframe computer offers beneficial speed and volume of data processing, but offers poor usability to the users, who must then rely on computer personnel for tabulation of the data into a useable form. Thus the placement of the data into a useful tabulated form in this processing method can be highly time consuming, and is not performed in real-time.

Personal computers, on the other hand, can offer good usability through the use of Graphical User Interfaces and the like, but are as yet unable to match the processing power of the mainframe computer. Data tabulation tools on personal computers can typically only handle low volumes of data, or low speed of tabulation.

Summary of the Invention

In accordance with the present invention, there is provided a method for data tabulation processing of a data file having a plurality of records in a plurality of data fields, comprising:

i) a pre-processing stage in which, for each individual data field, each distinct

data value is identified and allocated a numerical identifier unique for that field; and

ii) a tabulation stage in which, for each data record, a cell of a result array is determined based on the numerical identifiers for that record, and the result array cell incremented.

Preferably, the pre-processing stage includes generating from said data file an encoded data file containing the numerical identifiers for the data values in each field, and a mapping file which stores a correspondence between each of the distinct data values in the fields and the corresponding numerical identifiers.

Preferably the tabulation stage includes selecting at least two data fields from the plurality of data fields for tabulation, and generating the result array utilising the numerical identifiers for the selected data fields. In a preferred form of the invention, for N selected data fields, a cell of the result array is identified for each data record according to:

$$I = K_1 + D_1K_2 + D_1D_2K_3 + \ldots + D_1D_2\ldots D_{N-1}K_N$$

where I is the cell identity,

$K_1, K_2, \ldots, K_N$ are the numerical identifiers for the record in the selected fields,

and $D_1, D_2, \ldots, D_{N-1}$ are numbers of distinct values in the selected fields.

The present invention also provides a system for data tabulation processing of a data file having a plurality of records in a plurality of data fields, comprising:

i) a coding processor in which, for each individual data field, each distinct data value is identified and allocated a numerical identifier unique for that field; and

ii) a tabulation processor in which, for each data record, a cell of a result array is determined based on the numerical identifiers for that record, and the result array cell incremented.

Embodiments of the invention allow high volume tabulation of data in personal computers with speed exceeding that previously offered by mainframe computers. High volume data tabulation, such as population census, can be achieved in near real time response.

Embodiments of the present invention can enable large amounts of data to be tabulated at very high speed, for example, on an Intel (TM) 80x86 based personal computer. The invention can be implemented, for example, in the form of a set of Remote Procedure Call functions that can be used in a client-server or stand-alone application. In tests performed, a 5-level tabulation of 3 million records of data was achieved in less than 40 seconds on a Intel Pentium-90MHz machine. The preferred form of the invention achieves this by processing a flat ASCII data file into an encoded file that allows fast collation of information. This preprocessing allows quick array mapping which in turn allows quick data tabulation.

Brief Description of the Drawings

A preferred embodiment of the present invention is described in detail hereinafter, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram illustrating a pre-processing subsystem of the preferred embodiment;

Figure 2 illustrates an example of an input file and an input description file;

Figure 3 is a block diagram illustrating a tabulation subsystem of the preferred embodiment;

Figure 4 illustrates the form of an encoded field file for N fields according to the preferred embodiment; and

Figure 5 illustrates the form of a tabulation result array.

Detailed Description of the Preferred Embodiment

Data tabulation is an important and well accepted tool in information analysis. Data tabulation involves processing a data file having a plurality of fields to count of the number of records for each combination of distinct field value for the multiple fields. Tabulation thus requires the computation of the number of records with a certain combination of distinct values for the specified fields. The result of a tabulation of a data file contains this number for all possible combinations of distinct values between the fields selected. An example of a simple data file and resulting tabulation is illustrated below in Tables 1 and 2.

ID	Sex	Area
1	M	A
2	M	B
3	F	B
4	F	B
5	M	B

Table 1: Sample data file

Sex/Area	A	B
M	1	2
F	0	2

Table 2: Sample tabulation of Sex v Area
Request tuples: (Sex), (Area)

Table 1 is an example of a simple data file of five records each having two primary data fields (Sex, Area) and an identifier field (ID). Table 2 is an example of a tabulation of the data from Table 1. As can be seen in Table 2, from the data tabulation it is possible to easily identify the number of data records from the data file which have a particular combination of attributes of each data field.

The preferred form of the present invention is embodied in a system that comprises two subsystems: pre-processing subsystem and a tabulation subsystem. The pre-processing subsystem transforms an input data file into an encoded file which enables fast tabulation of the data. The tabulation subsystem responds to a request for data tabulation on the input file and returns the results in a tabular form.

Referring to Figure 1, the processes involved in the pre-processing subsystem 2 of the data tabulation system of a preferred form of the invention are illustrated in a block diagram. Input is provided to the pre-processing subsystem 2 from an input data file 20 in the form of

an ASCII flat file that contains information equivalent to a usual database table. Figure 2 shows an example of such an input data file 20, in this case with 4 records each having three data fields 26. Each line in the file represents single record whereby individual fields are separated by a "tab" character. The input file is accompanied by an input description file 22 that contains information about the type and length of each field in the input file. An example of an input description file is also illustrated in Figure 2, which shows examples of the data field label 34, field type 36 and length 38.

The input file 20 and input description file 22 is provided to a field separator unit 10. The field separator produces, for each field in the input file, an individual field file 12 that contains the field value for all records in the input file 20 for that field. The records in the individual field file 12 are arranged in the same order as those in the input file 20. This separation is achieved by copying field values in records of the input file to the individual field file 12 with information provided by the input description file 22. Upon completion of field separation, the input file 20 can then discarded from the system.

All individual field files 12 having discrete fields (e.g. fields of type integer or string) are then submitted to a distinct code mapping unit 14. The distinct code mapping unit 14 reads through the individual field files 12, and for each individual field file assigns an incremental numerical code to all field values found therein. This numerical code starts from 0 and increments by 1 with each distinct field value found. That is, the first distinct field value found in the field will be assigned the code 0, the second distinct value 1 and so on. All the encoded records for each individual field file 12 are then copied into a corresponding encoded field file 16. The encoded file 16 contains the encoded field values in the same order as the individual field file 12. A mapping file 18 is also produced for each field to capture the code assigned for each field value.

The individual field files having continuous field types (e.g. fields of floating point or date type) are not subjected to any further processing.

The pre-processing stage ends when all fields in the input file 20 are extracted into individual field files and all discrete fields are further converted into corresponding encoded files 16. The encoded files 16 and individual field files 12 are then stored, for example on a computer disk for further processing upon a tabulation request.

Figure 4 diagrammatically illustrates encoded field files 16 for an input data file having M discrete fields and Q data records. Field 1 as shown contains D_1 distinct values (A(1), A(2), ..., A(D_1)), field 2 has D_2 distinct values (B(1), B(2), ..., B(D_2)), and so on up to field M which has D_M distinct values (E(1), E(2), ..., E(D_M)). Each of the Q data elements in each encoded data file is allocated a corresponding distinct numerical value.

Referring to Figure 3, the processes involved in the tabulation subsystem 30 of the data tabulation system of the preferred embodiment are illustrated in a block diagram. A tabulation request 40 and the encoded field files 16 are provided to a tabulation unit 42. The tabulation request specifies a plurality of field tuples. Each field tuple specifies a field of the data file. For example a tabulation request may typically specify a pair of field tuples, with one tuple representing the fields required for the columns and the other tuple representing those required for the rows of the result table.

Upon receiving the tabulation request, the tabulation unit formulates an empty result array in the form of a one-dimensional array for storing integer values. The cells in the result array are initialised to 0. The number cells in the array is determined by the product of the number of distinct values in each field specified in the tabulation request. An example of a result array 44 is represented diagrammatically in Figure 5, having a plurality of X cells 52. In this instance the tabulation request specifies fields P and Q, so that the number of cells in the result array is computed by $X = D_P * D_Q$. Each cell in the result array is used to store the number of records for one combination of field values. The mapping of a cell to a particular combination is achieved through an algorithm outlined below. The algorithm is represented in its generalised form for cross-tabulation of N fields, and it will be appreciated that the integer N may in fact be any number from 2 up to the total number of fields in the data file.

Algorithm for extracting result values for result array for cross tabulation of N fields each with distinct values D_1, D_2, ... , D_N, respectively.

a) Obtain codes for the field values from the encoded field files.

b) Let the codes for the N field values, one from each field, be represented by K_1, K_2, ..., K_N.

The index, I, of the cell in the result array for each data record corresponding to the intersection of the specified field values is then given by:

$$\mathbf{I = K_1 + D_1K_2 + D_1D_2K_3 + ... + D_1D_2...D_{N-1}K_N} \qquad \textbf{(1)}$$

The tabulation result array 44 can then be completed by:

i) for each data record, systematically retrieving from the corresponding encoded field files 16 the distinct value codes from the fields specified in the tabulation request;

ii) computing the result array index I utilising equation (1) above; and

iii) incrementing the value in the result array cell identified by the computed index.

When all of the records from the relevant fields have been read and processed in this way, the result array will contain the tabulation results. The tabulation results may then be easily represented in a graphical tabular form, for example, with the rows and columns and contextual meaning of the result array cells identified with reference to the map files 18.

It will be readily recognised by those of ordinary skill in the art that the described embodiment of the present invention can be easily implemented on any suitable form of digital processing apparatus, such as a personal desktop computer or portable computer, or a mainframe computer if desired. In that case, the invention would be implemented by way of computer instruction codes for controlling the computing machinery, and the code may be written in any desired language, and stored in any desired format and medium, as would be apparent to those skilled in the art. The described pre-processing and data tabulation procedures allow for relatively fast tabulation of desired data fields. In one test, tabulation of 3 million records at 5 levels was achieved in 40 seconds on a Intel Pentium-90 (TM)

computing machine.

The terms and expressions employed herein are used as terms of description and not of limitation, and there is no intention, in use of such terms and expressions, of excluding any equivalents of the features shown and described or portions thereof, but it is recognised that various modifications are possible within the scope of the method and system claimed.